POCKET COMPUTER NEWSLETTER

© Copyright 1982 — Issue 15 May

GENERATING THE CHROMATIC SCALE ON THE SHARP PC-1500

As indicated on pages 84 and 85 of the *Sharp PC-1500 Instruction Manual* the BEEP statement allows the programmer to create tones for game playing, produce warning signals and serve other annunciator purposes. Three parameters specified as suffixes to the statement permit one to specify the number, frequency and duration of the tones emitted. Theoretically one should be able to play music. But what are the specifications needed to produce each tone on the chromatic scale?

Brian Peterson, 6807 N. Sheridan Rd. – Apt. 520, Chicago, IL 60626 has figured this information out and provided it for *PCN* readers.

Brian has also learned that the duration of a BEEP command varies with the frequency. The duration of all the tones can be made approximately equal by using the following relationship:

$$L=10^{(N/48)}$$

In this formula L represents the length of the tone and N represents the number of the tone as shown in the accompanying table. The last column in the table shows the calculated value for L using this formula.

Table *PC-1500 Chromatic Scale Using BEEP Function*

#	KY	TONE	L VAL	26	A#	55	3.48
1	A	255	1.04	27	B	52	3.65
2	A#	244	1.10	28	C	49	3.83
3	B	229	1.15	29	C#	45	4.01
4	C	219	1.21	30	D	43	4.21
5	C#	203	1.27	31	D#	41	4.42
6	D	195	1.33	32	E	37	4.64
7	D#	184	1.39	33	F	35	4.86
8	E	172	1.46	34	F#	32	5.10
9	F	163	1.53	35	G	30	5.36
10	F#	151	1.61	36	G#	28	5.62
11	G	143	1.69	37	A	26	5.89
12	G#	133	1.77	38	A#	24	6.18
13	A	125	1.86	39	B	22	6.49
14	A#	118	1.95	40	C	21	6.81
15	B	111	2.05	41	C#	19	7.14
16	C	105	2.15	42	D	18	7.49
17	C#	98	2.26	43	D#	17	7.86
18	D	94	2.37	44	E	15	8.25
19	D#	89	2.48	45	F	14	8.65
20	E	81	2.61	46	F#	12	9.08
21	F	77	2.73	47	G	11	9.53
22	F#	72	2.87	48	G#	10	10.00
23	G	68	3.01	49	A	9	10.49
24	G#	64	3.16	50	A#	8	11.00
25	A	59	3.31	51	B	7	11.54

Multiply this value by 10 to produce a note of short duration, 20 to get a longer note, and so forth.

By the way, middle C in the accompanying table is note number 28 (having a tone value of 49 and L value of 3.83).

Notes with tone values of less than 14 (notes numbered 46 and above in the table) are slightly out of tune. They may still be satisfactory in certain applications.

If you want to try your hand at playing a little music, load in the program shown in the accompanying listing and have at it on the number keys.

Brian recently sent in a striking rendition of Bach's *Prelude Number II* from *The Well Tempered Clavier*. He is willing to write a tutorial article on how to convert melodies to play on the PC-1500. Let us know if this strikes your fancy.

Program *Musical Keyboard for the Sharp PC-1500*

```
100: X$=INKEY$ : IF
     X$="" THEN 100
110: IF (X$<"0")+(X
     $>"9") THEN 100
120: X=(ASC (X$)-47
     )*10+190: GOTO
     X
200: BEEP 1,52,72:
     GOTO 100
210: BEEP 1,49,76:
     GOTO 100
220: BEEP 1,43,84:
     GOTO 100
230: BEEP 1,37,92:
     GOTO 100
240: BEEP 1,35,98:
     GOTO 100
250: BEEP 1,30,104:
     GOTO 100
260: BEEP 1,26,118:
     GOTO 100
270: BEEP 1,22,130:
     GOTO 100
280: BEEP 1,21,136:
     GOTO 100
290: BEEP 1,18,150:
     GOTO 100

STATUS 1
                274
```

UNDERSTANDING THE SHARP PC-1500

This is the first of a series of columns being prepared by *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158,* with the object of the series being to help newcomers get acquainted with their PCs.

Hexadecimal Numbers

People ordinarily write numbers in *decimal* form. These numbers have a base of 10. The value 726 is understood to mean a total of 7 hundreds plus 2 tens plus 6 ones. Stated another way, it equals $7*10^2 + 2*10^1 + 6*1$.

Computers use numbers based on the *binary* system which only has two digits, 0 and 1. It turns out that people can deal with groups of binary digits much more efficiently if they are grouped together and considered as numbers written with a base of 16. In this case, each of the digits is understood to be multiplied by a power of 16. Numbers in this format are referred to as *hexadecimal* numbers. To illustrate, the hexadecimal number 643 has a total value of $6*16^2 + 4*16^1 + 3*1$. If you multiply this out, you will find that this is equivalent to the decimal (base 10) value 1603.

Decimal numbers are written using ten different digits (0, 1, 2, 3, . . . , 9). The hexadecimal system requires the use of 16 different digits. The letters A through F are used to represent the "digits" ten through fifteen. Thus, the hexadecimal digits are 0, 1, 2, . . . , 9, A, B, C, D, E and F. The hexadecimal number 2B represents $2*16^1 + 11*1$, which is 43 decimal.

Using the ampersand (&) on the Sharp PC-1500 as a prefix indicates that the number is to be interpreted as hexadecimal. *PCN* will use the same convention. Thus, &643 is understood to be the hexadecimal notation of 1603 (decimal).

Just to convince yourself, enter &643 on your PC-1500 and press ENTER. It will respond with the decimal equivalent: 1603.

Peeking at Memory

The BASIC interpreter in the PC-1500 contains a number of instructions that are not mentioned in the instruction manual. One of these is the PEEK function. It can be used to see what is stored at any location in memory. Among other uses, PEEK makes it comparatively easy to discover how the unit's memory is utilized.

To learn how PEEK works, try this example. First, execute the LOCK statement using the keyboard. This directive effectively locks the computer in its present mode of operation. Next, type in PEEK 31231 or PEEK &79FF and then press the ENTER key. The display will show the value 0. Now, execute the UNLOCK statement. Once again type in PEEK 31231 and press ENTER. Note that now the value 96 is displayed. What is going on?

The memory address 31231 (&79FF) is being used by the computer to store the information that the current mode is either locked or unlocked!

Although PEEK returns values in decimal notation, the computer can be considered to be storing numbers in hexadecimal format. The computer also refers to memory addresses as hexadecimal values. The decimal address 31231 is &79FF in hexadecimal and the decimal value 96 is &60. Speaking in hexadecimal terms, location &79FF holds &60 when the PC-1500 is locked and &0 when it is unlocked.

Each memory location is capable of storing a two-digit hexadecimal number ranging from &00 through &FF. A hexadecimal digit can represent the value of a group of 4 binary digits. Two hexadecimal digits can thus represent 8 binary digits. A group of eight binary digits in a computer is often referred to as a *byte.* So, two hexadecimal digits are needed to represent each byte of information in the PC-1500.

Taking a Look at Variable Storage

By doing a lot of PEEKing operations I have been able to ascertain where the variable values are stored in the PC-1500. The variable A is stored in memory locations &7900 through &7907. Notice that this amounts to eight bytes of storage. To experiment yourself, set the variable A to the value 74.685. Then PEEK at the eight locations &7900 through &7907. You should obtain the following results:

Location	PEEK Result (Decimal)	Actual Storage (Hexadecimal)
&7900	1	01
&7901	0	00
&7902	116	74
&7903	104	68
&7904	80	50
&7905	0	00
&7906	0	00
&7907	0	00

The value 01 stored in location &7900 is the exponent of ten for the stored number. (74.685 is stored as 7.4685E1.) Address &7901 is used to store the sign of the number. (Try repeating the above with a negative number and you will see how a negative sign is stored.) Note that (exclusive of the exponent) only the digits zero to nine are used. That is, the digits of the number are stored in decimal form. (Computers often store numbers in strictly binary or hexadecimal form.) However, if you do some experimenting, you will find that the exponent of a stored number uses hexadecimal notation. Furthermore, if the exponent is negative, it is stored as the sum of the negative exponent and the number 256. For instance, an exponent of -18, added to 256 yields a value of 238. When this is converted to hexadecimal the result is &EE. Of course, since PEEK converts to decimal automatically, you will simply see the value 238 when you PEEK at the address holding the exponent!

The string variable A$ is stored in locations &78C0 through &78CF, allowing room for storage of 16 bytes. Each character of the stored string is stored as a single byte. The stored number for each character is the ASCII code for that character. The ASCII codes are tabulated in Appendix C of the PC-1500's Instruction Manual. Try PEEKing at a stored string variable.

Program *Sharp PC-1500 Memory Dump to LCD*

```
10 INPUT "STARTING LOCATION? ";A
20 A$="0123456789ABCDEF"
30 WAIT 0:FOR B=0 TO 7:C=PEEK(A+B):D=1+INT(C/16)
40 PRINT MID$(A$,D,1);MID$(A$,C+17-16*D,1);" ";
   :NEXT B
50 WAIT:PRINT:GOTO 10
```

If you would like to do some exploring on your own and you are getting tired of typing in the command PEEK over and over, the accompanying short program may be used to display eight stored numbers in a row, all neatly converted to hexadecimal. *[Another "memory dump" program elsewhere in this issue may be used to explore large areas of memory with output to the printer unit. —N.W.]*

To use the memory display program, respond to the prompt by giving the memory address where you want to start examining a series of 8 bytes. You can also easily modify the program to automatically advance to the next 8 bytes by changing line 50 of the program to read:

50 WAIT:PRINT:A=A+8:GOTO 30

Next issue I plan to discuss the POKE directive which allows you to place values directly into memory locations. I'll have other goodies on the subject of understanding and utilizing the PC-1500 too!

PEEKING IN THE PC-1500

It didn't take users long to discover that there are a number of capabilities of the Sharp PC-1500 that are not mentioned in the instruction manual. Undocumented directives include PEEK, POKE, CALL and OPN. *George Fergus* was one of the first users to get a report in to *PCN* listing some of his discoveries using the PEEK instruction. This command permits one to examine the contents of memory by giving the directive: PEEK X, where X represents an address in decimal or hexadecimal (indicated by preceeding the value with the "&" sign) format. Some of the information George has gleaned, such as the locations (addresses) of RAM elements in the basic unit and the tokens used in the BASIC interpreter, is shown in accompanying tables.

If you want to do some exploring on your own, an accompanying routine provides a means of "dumping" portions of memory using the CE-150 printer. In addition to providing a hexadecimal dump, the program prints ASCII-equivalent characters for appropriate values underneath each line of hexadecimal code. Examining memory starting at hexadecimal address C000 (49152 decimal) reveals the coding of the BASIC interpreter. If you use the dump to examine the first few hundred bytes of ROM code you will discover the token conversion table used by the interpreter.

For those of you who may not have had experience with the PEEK directive, *Norlin Rober* provides an introduction to the subject in an article elsewhere in this issue.

Table *PC-1500 BASIC Token Codes*

F1 50	AND	F1 81	ARUN	F1 B1	TO
F1 51	OR	F1 82	BEEP	F1 B3	WAIT
F1 58	MEM	F1 83	CONT	F1 B4	ERROR
F1 5B	TIME	F1 86	GRAD	F1 B5	LOCK
F1 5C	INKEY$	F1 87	CLEAR	F1 B6	UNLOCK
F1 5D	PI	F1 8A	CALL		
F1 60	ASC	F1 8B	DIM	F0 84	CURSOR
F1 61	STR$	F1 8C	DEGREE	F0 85	USING
F1 62	VAL	F1 8D	DATA	F0 88	CLS
F1 63	CHR$	F1 8E	END	F0 89	CLOAD
F1 64	LEN	F1 92	GOTO	F0 8F	MERGE
F1 65	DEG	F1 94	GOSUB	F0 90	LIST
F1 66	DMS	F1 96	IF	F0 91	INPUT
F1 67	STATUS	F1 98	LET	F0 93	GCURSOR
F1 68	POINT	F1 99	RETURN	F0 95	CSAVE
F1 6B	SQR	F1 9A	NEXT	F0 97	PRINT
F1 6D	NOT	F1 9B	NEW	F0 9F	GPRINT
F1 6E	PEEK#	F1 9C	ON	F0 B2	CHAIN
F1 6F	PEEK	F1 9D	OPN	F0 B5	COLOR
F1 70	ABS	F1 9E	OFF	F0 B6	LF
F1 71	INT	F1 A0	POKE#	F0 B7	LINE
F1 72	RIGHT$	F1 A1	POKE	F0 B8	LLIST
F1 73	ASN	F1 A2	PAUSE	F0 B9	LPRINT
F1 74	ACS	F1 A3	P	F0 BA	RLINE
F1 75	ATN	F1 A4	RUN	F0 BB	TAB
F1 76	LN	F1 A5	FOR	F0 BC	TEST
F1 77	LOG	F1 A6	READ		
F1 78	EXP	F1 A7	RESTORE	E7 A9	RMT
F1 79	SGN	F1 A8	RANDOM		
F1 7A	LEFT$	F1 AA	RADIAN	E6 80	CSIZE
F1 7B	MID$	F1 AB	REM	E6 81	GRAPH
F1 7C	RND	F1 AC	STOP	E6 82	GLCURSOR
F1 7D	SIN	F1 AD	STEP	E6 83	LCURSOR
F1 7E	COS	F1 AE	THEN	E6 84	SORGN
F1 7F	TAN	F1 AF	TRON	E6 85	ROTATE
F1 80	AREAD	F1 B0	TROFF	E6 86	TEXT

Table *PC-1500 Partial Memory Map*

&4008	–	40C3	Reserve Program Area (188 bytes)
&40C6	–	47FF	User Main Program Area (1850 bytes)
&7050	–	70FF	Fixed Variables E$ – O$ (11 x 16 bytes)
&7150	–	71FF	Fixed Variables P$ – Z$ (11 x 16 bytes)
&78C0	–	78FF	Fixed Variables A$ – D$ (4 x 16 bytes)
&7900	–	79CF	Fixed Variables A – Z (26 x 8 bytes)
&C000	–	FFFF	BASIC Interpreter (16K bytes in ROM)

Program *PC-1500 Hexadecimal Memory Dump*

```
10:CSIZE 1:INPUT
   "STARTING ADDR
   ESS? ";A
20:A$="0123456789
   ABCDEF"
25:WAIT 0:W=INT (
   A/4096)
26:X=INT ((A-(W*4
   096))/256)
27:Y=INT ((A-(W*4
   096)-(X*256))/
   16)
28:Z=INT ((A-(W*4
   096)-(X*256)-(
   Y*16)))
29:LPRINT MID$ (A
   $,W+1,1);MID$
   (A$,X+1,1);
   MID$ (A$,Y+1,1
   );MID$ (A$,Z+1
   ,1);"  ";
30:WAIT 0:FOR B=0
   TO 7:C=(PEEK (
   A+B))AND (&F0)
   :D=1+INT (C/16
   ):E=(PEEK (A+B
   ))AND (&0F)
40:LPRINT MID$ (A
   $,D,1);MID$ (A
   $,E+1,1);" ";:
   NEXT B
45:LPRINT :LPRINT
   "        ";:FOR
   B=0TO 7:LPRINT
   (CHR$ (PEEK (A
   +B)));"  ";:
   NEXT B:LPRINT
   :LF 1
50:A=A+8:GOTO 25
```

Table *PC-1500 Key Memory Locations*

&7600	–	764D	Display memory
&7700	–	774D	Display Memory
&764E	–	764F	RUN/PRO/RESERVE settings, trig mode, SHIFT status, SML status
&7800	–	78BF	Memory pointers, format settings
&79D0	–	79FE	Settings primarily involving the CE-150
&79FF			LOCK/UNLOCK status
&7A00	–	7AFF	Calculation registers, data stack, FOR/NEXT stack, subroutine stack, etcetera
&7B00	–	7B07	Random number storage
&7B08	–	7B0F	Counter, used for automatic 7-minute shutoff, possibly other purposes
&7B10	–	7B5F	String buffer
&7B60	–	7BAF	Data to be outputted to display
&7BB0	–	7BFF	Input buffer
&A000	–	BFFF	ROM in the CE-150

CURVE FITTING

The accompanying program will fit a series of X-Y data points to all of the following curves: linear, exponential, logarithmic and power. It is a good example of how a lot of practical computing power can be packed into a single program on a pocket computer. The program was crafted by: *Thomas S. Cox, 102 Evergreen Street, Easley, SC 29640.*

After loading the program, place the PC in the DEF mode.

Initialization

Use SHIFT/Space to initialize the program and/or to add additional data points. Answer appropriately to the initialization query. Enter X and Y data for each point. Press ENTER after reviewing data for correctness.

Deleting/Correcting Data

Press SHIFT/G to change or delete data points. Respond appropriately to prompts. Note that SHIFT/G must be used each time a data point is to be altered.

Obtaining Best Fit

Press SHIFT/F to determine which of the four curves yields the highest R^2 value. Information concerning the best fit is output in the sequence: 1. Curve type and descriptive equation, 2. Constant term A, 3. Variable term B, 4. Correlation coefficient R, 5. Coefficient of determination R^2.

Specific Curves

To view the parameters for any of the four curves: Press SHIFT/L for linear, SHIFT/A for exponential, SHIFT/S for logarithmic and SHIFT/D for the power curve. The parameter data is output in the same sequence as shown for the best fit curve.

Estimate of X or Y

To obtain an estimate of an X or Y value *for the currently selected curve,* press SHIFT/K (for X) or SHIFT/J (for Y). Enter the appropriate coordinate value when prompted. The program then calculates the predicted value.

Restrictions

If X and/or Y are negative the best fit procedure is skipped. The message "X OR Y OR BOTH NEG" is displayed. Use manual curve selection to obtain a fit. If not feasible, the same "X OR Y OR BOTH NEG" message will be shown.

Tight Fit

The version shown for the Sharp PC-1211/Radio Shack PC-1 will only have 2 program steps left when loaded with no extra spaces as shown in the listing. Variables A – Z are utilized by the program.

Program *Curve Fitting (Sharp PC-1211/Radio Shack PC-1 Version)*

```
 1 " "USING:
   PRINT"CURVE FIT-V4":
   INPUT"CLR (Y/N)";Z$:
   IF Z$="Y"CLEAR
 2 C=1:
   Z=N+1:
   PAUSE"PT#";Z:
   INPUT"X=";L:
   IF L<=0 LET J=1
 3 X=L:
   INPUT"Y=";Y:
   IF Y<=0 LET T=1
 4 GOSUB 5:
   BEEP 2:
   PRINT"X";N;"=";X;" Y";N;"=";Y:
   GOTO 2
 5 D=D+CX:
   E=E+CXX:
   F=F+CY:
   G=G+CYY:
   H=H+CXY:
   N=N+1*C:
   IF J*T=1 RETURN
 6 IF J<>1 LET M=M+C*LN X:
   O=O+C*LN X*LN X:
   K=K+C*Y*LN X
 7 IF T<>1 LET P=P+C*LN Y:
   Q=Q+C*LN Y*LN Y:
   I=I+CX*LN Y
 8 IF J+T=0 LET S=S+C*LN X*LN Y
 9 RETURN
10 "L"IF W=1 THEN 14
11 U=1:
   PRINT"LIN. Y=B*X+A":
12 B=(H-DF/N)/(E-DD/N):
   A=(F/N-BD/N)
14 R=(NH-DF)/SQR((NE-DD)*(NG-FF)):
   IF W=1 RETURN
16 GOTO 96
20 "A"IF W=1 THEN 24
21 U=2:
   PRINT"EXP. Y=A*EXP(B*X):
   IF T=1 GOTO 61
22 B=(I-PD/N)/(E-DD/N):
   A=EXP (P/N-BD/N)
24 R=(NI-DP)/SQR((NE-DD)*(NQ-PP)):
   IF W=1 RETURN
26 GOTO 96
30 "S"IF W=1 THEN 34
31 U=3:
   PRINT"LOG. Y=B*LNX+A":
   IF J=1 GOTO 61
32 B=(K-MF/N)/(O-MM/N):
   A=(F-BM)/N
34 R=(NK-FM)/SQR((NO-MM)*(NG-FF)):
   IF W=1 RETURN
36 GOTO 96
40 "D"IF W=1 THEN 44
41 U=4:
   PRINT"PWR. Y=A*X^B":
   IF T+J>0 GOTO 61
42 B=(S-MP/N)/(O-MM/N):
   A=EXP (P/N-(BM/N))
44 R=(NS-MP)/SQR((NO-MM)*(NQ-PP)):
   IF W=1 RETURN
46 GOTO 96
48 "F"IF J+T>0THEN 61
50 W=1:
   V=0:
   U=0:
   FOR Z=1 TO 4:
   GOSUB (Z*10):
   C=RR:
   IF C>V GOSUB99
55 NEXT Z:
   USING:
   W=0:
   BEEP3:
   PRINT"BEST FIT WAS #";U
59 GOTO (U*10)
60 "K"PRINT"ESTIMATE OF X":
   INPUT"Y=";V:
   GOTO (U+61)
61 BEEP 1:
   PRINT"X OR Y OR BOTH NEG.":
   END
62 L=(V-A)/B:
   GOTO 97
63 L=(LNV-LNA)/B:
   GOTO 97
64 L=EXP((V-A)/B):
   GOTO 97
65 L=EXP ((LNV-LNA)/B):
   GOTO 97
67 USING"#####.###":
   RETURN
70 "G"BEEP 1:
   PRINT"DELETE":
   INPUT"DELETE LAST PT (Y/N)";Z$:
   IF Z$="Y"THEN 75
71 PRINT"PT# ";N:
   INPUT"X=";L:
   IF L<=0 LET J=1
72 X=L:
   INPUT"Y=";Y:
   GOSUB 98:
   GOTO 77
75 PRINT"X=";X;" Y=";Y:
   INPUT"OK TO DEL(Y/N)";Z$
76 IF Z$="N"THEN 71
77 C=-1:
   IF Y<=0 LET T=1
78 GOSUB 5:
   END
82 "J"PRINT "ESTIMATE OF Y":
   INPUT"X=";L:
   GOTO (U+85)
86 V=BL+A:
   GOTO 97
87 V=A*EXP (BL):
   GOTO 97
88 V=B*LNL+A:
   GOTO 97
89 V=A*L^B:
   GOTO 97
96 GOSUB 67:
   PRINT"A=";A:
   PRINT"B=";B:
   PRINT"R=";R:
   C=RR:
   PRINT"R^2=";C:
   END
97 PRINT"X=";L;" Y=";V:
   END
98 PRINT"X=";X;" Y=";Y:
   RETURN
99 U=Z:
   V=C:
   RETURN
```

PC-1500 Version

Implied multiplication, used extensively to fit the program in a PC-1, cannot be used in a Radio Shack PC-2/Sharp PC-1500. However, the additional memory in these models allows the program to fit with room to spare when implied directives are changed to explicit representations.

An accompanying listing illustrates how the program appears on a PC-1500/PC-1. Ordinarily, *PCN* readers will be left on their own to adapt PC-1211/PC-1 programs to PC-1500/PC-2 units. This might typically involve watching out for implied multiplication as well as dropped right hand parens, ending quotation marks, etc.

Program *Curve Fitting (Sharp PC-1500/Radio Shack PC-2 Version)*

```
1:" "USING :
  PRINT "CURVE F
  IT-U4":INPUT "
  CLR (Y/N)";Z$:
  IF Z$="Y"CLEAR
2:C=1:Z=N+1:
  PAUSE "PT#";Z:
  INPUT "X=";L:
  IF L<=0LET J=1
3:X=L:INPUT "Y="
  ;Y:IF Y<=0LET
  T=1
4:GOSUB 5:BEEP 2
  :PRINT "X";N;"
  =";X;" Y";N;"=
  ";Y:GOTO 2
5:D=D+C*X:E=E+C*
  X*X:F=F+C*Y:G=
  G+C*Y*Y:H=H+C*
  X*Y:N=N+1*C:IF
  J*T=1RETURN
6:IF J<>1LET M=M
  +C*LN X:O=O+C*
  LN X*LN X:K=K+
  C*Y*LN X
7:IF T<>1LET P=P
  +C*LN Y:Q=Q+C*
  LN Y*LN Y:I=I+
  C*X*LN Y
8:IF J+T=0LET S=
  S+C*LN X*LN Y
9:RETURN
10:"L"IF W=1THEN
   14
11:U=1:PRINT "LIN
   . Y=B*X+A"
12:B=(H-D*F/N)/(E
   -D*D/N):A=(F/N
   -B*D/N)
14:R=(N*H-D*F)/√(
   (N*E-D*D)*(N*G
   -F*F)):IF W=1
   RETURN
16:GOTO 96
20:"A"IF W=1THEN
   24
21:U=2:PRINT "EXP
   . Y=A*EXP(B*X)
   ":IF T=1GOTO 6
   1
22:B=(I-P*D/N)/(E
   -D*D/N):A=EXP
   (P/N-B*D/N)
24:R=(N*I-D*P)/√(
   (N*E-D*D)*(N*Q
   -P*P)):IF W=1
   RETURN
26:GOTO 96
30:"S"IF W=1THEN
   34
31:U=3:PRINT "LOG
   . Y=B*LNX+A":
   IF J=1GOTO 61
32:B=(K-M*F/N)/(O
   -M*M/N):A=(F-B
   *M)/N
34:R=(N*K-F*M)/√(
   (N*O-M*M)*(N*G
   -F*F)):IF W=1
   RETURN
36:GOTO 96
40:"D"IF W=1THEN
   44
41:U=4:PRINT "PWR
   . Y=A*X^B":IF
   T+J>0GOTO 61
42:B=(S-M*P/N)/(O
   -M*M/N):A=EXP
   (P/N-(B*M/N))
44:R=(N*S-M*P)/√(
   (N*O-M*M)*(N*Q
   -P*P)):IF W=1
   RETURN
46:GOTO 96
48:"F"IF J+T>0
   THEN 61
50:W=1:V=0:U=0:
   FOR Z=1TO 4:
   GOSUB (Z*10):C
   =R*R:IF C>V
   GOSUB 99
55:NEXT Z:USING :
   W=0:BEEP 3:
   PRINT "BEST FI
   T WAS #";U
59:GOTO (U*10)
60:"K"PRINT "ESTI
   MATE OF X":
   INPUT "Y=";V:
   GOTO (U+61)
61:BEEP 1:PRINT "
   X OR Y OR BOTH
    NEG.":END
62:L=(V-A)/B:GOTO
   97
63:L=(LN V-LN A)/
   B:GOTO 97
64:L=EXP ((V-A)/B
   ):GOTO 97
65:L=EXP ((LN V-
   LN A)/B):GOTO
   97
67:USING "######.#
   ##":RETURN
70:"G"BEEP 1:
   PRINT "DELETE"
   :INPUT "DELETE
    LAST PT (Y/N)
   ";Z$:IF Z$="Y"
   THEN 75
71:PRINT "PT# ";N
   :INPUT "X=";L:
   IF L<=0LET J=1
72:X=L:INPUT "Y="
   ;Y:GOSUB 98:
   GOTO 77
75:PRINT "X=";X;"
    Y=";Y:INPUT "
   OK TO DEL(Y/N)
   ";Z$
76:IF Z$="N"THEN
   71
77:C=-1:IF Y<=0
   LET T=1
78:GOSUB 5:END
82:"J"PRINT "ESTI
   MATE OF Y":
   INPUT "X=";L:
   GOTO (U+85)
86:V=B*L+A:GOTO 9
   7
87:V=A*EXP (B*L):
   GOTO 97
88:V=B*LN L+A:
   GOTO 97
89:V=A*L^B:GOTO 9
   7
96:GOSUB 67:PRINT
   "A=";A:PRINT "
   B=";B:PRINT "R
   =";R:C=R*R:
   PRINT "R^2=";C
   :END
97:PRINT "X=";L;"
    Y=";V:END
98:PRINT "X=";X;"
    Y=";Y:RETURN
99:U=Z:V=C:RETURN

STATUS 1
                1677
```

STATUS 1 at end of listing indicates the number of bytes utilized.

PUT THOSE SOFTKEYS TO WORK FOR YOU!

As *George Fergus* was quick to discover, you can quickly set up the Sharp PC-1500 to save yourself a lot of keying when developing programs. Just assign the most commonly used punctuation symbols that normally require two keystrokes (SHIFT and then the desired character) to a group of softkeys. For instance, using all six keys from left to right, you could define them to represent the symbols:

: ; , ? # $

Use the RCL key too, to display what you have assigned to each softkey to assist in remembering your assignments. The technique sure saves a lot of work.

LCD GRAPHICS FOR THE SHARP PC-1500

David G. Motto, 3639 Roosevelt Circle, Jackson, MI 49203, sent in the codes shown in the accompanying listing for producing a variety of graphics on the Sharp PC-1500 liquid crystal display.

The codes are presented within DATA statements. A simple routine at the start of the listing (lines 1 – 100) may be used to access the data statements and display the various patterns. Naturally, you can extend the concepts to produce your own new patterns and incorporate them in your own programs. Note the complete set of chess pieces provided by Dave. Does that give anyone ideas? See pages 91 and 92 of the *Sharp PC-1500 Instruction Manual* for details on producing LCD graphics.

Program *LCD Graphics for the Sharp PC-1500 and Radio Shack PC-2.*

```
  1:RESTORE 1000
  5:DIM X$(0)*80
100:CLS :WAIT 100:
    GCURSOR 0:READ
    X$(0):GPRINT X
    $(0):GOTO 100
1000:DATA "081436
     49361408"
1010:DATA "776355
     08556377"
1020:DATA "553677
     00773655"
1030:DATA "1C0055
     4955001C"
1040:DATA "080014
     002A0055"
1050:DATA "775577
     00775577"
1060:DATA "7F2214
     0814227F"
1070:DATA "63411C
     141C4163"
1080:DATA "493622
     49223649"
1090:DATA "775D77
     22775D77"
1100:DATA "364949
     36494936"
1110:DATA "7F415D
     555D417F"
1120:DATA "1C2A49
     7F492A1C"
1200:DATA "0E1121
     4221110E":
     REM   HEART
1210:DATA "081C4A
     7F4A1C08":
     REM   CLUB
1220:DATA "081422
     41221408":
     REM   DIAMON
     D
1230:DATA "183C1E
     7F1E3C18":
     REM   SPADE
1300:DATA "1D2222
     4122221D":
     REM   TAURUS
1310:DATA "21322C
     20207C20":
     REM   JUPITE
     R
1320:DATA "062979
     2906":REM
     FEMALE
1330:DATA "304848
     48350307":
     REM   MALE
1340:DATA "434C34
     2A161961":
     REM   SPIRAL
1350:DATA "04324A
     14292610":
     REM   SPIRAL
1360:DATA "1C3E7F
     7F7F3E1C":
     REM   BALL
1370:DATA "1C2241
     4141221C":
     REM   CIRCLE
1380:DATA "63594F
     454F5963":
     REM   BELL
1400:DATA "0E7C7E
     7C0E":REM
     ROOK (B)
1410:DATA "080C4E
     677E":REM
     KNIGHT (B)
1420:DATA "4C5E3F
     5E4C":REM
     BISHOP (B)
1430:DATA "427C7F
     7C42":REM
     QUEEN (B)
1440:DATA "407A7F
     7A40":REM
     KING (B)
1450:DATA "1C1E1F
     1E1C":REM
     PAWN (B)
1500:DATA "0E7C42
     7C0E":REM
     ROOK (W)
1510:DATA "080C4A
     257E":REM
     KNIGHT (W)
1520:DATA "4C5233
     524C":REM
     BISHOP (W)
1530:DATA "427C43
     7C42":REM
     QUEEN (W)
1540:DATA "407A4F
     7A40":REM
     KING (W)
1550:DATA "1C1211
     121C":REM
     PAWN (W)
1600:DATA "3F0B7F
     1F1F1F7F":
     REM   ELEPHA
     NT
1610:DATA "40787E
     7F7E7840":
     REM   BUILDI
     NG
1620:DATA "3F7F79
     79393F3F3838
     787C7C3838":
     REM   LOCOMO
     TIVE
1630:DATA "3C7C7C
     3C3C7C7C3C10
     ":REM   BOXC
     AR
1640:DATA "020302
     1C0C5C7C6C08
     10204040":
     REM   KANGAR
     OO
1650:DATA "010101
     0909191F1919
     18181C1E1A06
     060706060202
     02":REM   EN
     TERPRISE
1660:DATA "1A1E1F
     1C0C04040404
     04040E0E0E6F
     6F7F7B7B6860
     60":REM   KL
     INGON
1670:DATA "081412
     122448102040
     000000402010
     102040000000
     402010102040
     0000004020"
```

FROM THE WATCH POCKET

This column will be somewhat longer than usual. I have been developing a list of matters to discuss as a result of inquiries, industry developments and my own findings. So, here goes . . .

The Panasonic/Quasar HHCs

I have been looking over a Panasonic HHC lately, equipped with 4K of user memory in the basic unit plus a 8K external module. Do you think that means I have a "12K" system? Not really, the memory is "paged" or partitioned so that you are either using the 4K section or the 8K block.

I have spent about 35 – 40 hours actually operating the unit, much of that devoted to keying in BASIC programs and running some evaluations and benchmarks.

The application programs supplied with the unit, such as the time-keeping and appointment system and the personal filing system, are functional and easy to operate.

But I am somewhat disappointed with the BASIC operating system. Panasonic, in my opinion, would be wise to study Sharp's approach in this area, starting right with the editing capabilities.

A feature I find particularly annoying on the Panasonic when listing a BASIC program is the following: on long lines the display automatically starts scrolling to the end of the line. It has been my experience when scanning program listings on a PC that I can generally follow the flow of a program by reviewing the beginning of each line (rather than the end of lines). The extra key pressing needed on the Panasonic to stop the across-the-line-scrolling and get back to the start of the line or merely abort the display to get to the next line is inconvenient.

If you edit a line on the Panasonic you must press the ENTER key to have the unit accept the changes. However, as soon as this is done, the line disappears from the display. To check the line again, often desirable after alterations have been made, you must re-enter the entire LIST directive with the desired line number.

Also, when reviewing a program, while it is possible to roll the display from one line to the next in ascending line number order, you cannot roll back. Once you have gone beyond a line, the only way to get back to it is to abort the listing and enter a new LIST directive.

Keying in, editing and debugging BASIC programs, based on my experience, requires considerably more work than on a Sharp PC-1500. There are several factors that combine to produce this effect.

First, I find that the lack of a numeric/operand keypad slows down proceedings. All mathematical operators such as +, *, /, etc., require the pressing of the shift key and then the appropriate symbol. This is also the case for other commonly used characters such as the colon and semi-colon. The extra keystrokes make a difference when keying in long programs. Being able to quickly set up softkeys to reduce the keystrokes in this situation (as can be done on the Sharp PC-1500) would be a big boon.

The Panasonic keyboard normally operates in the lower case mode. BASIC statements and expressions are automatically converted to upper case when listed by the system – except when you have enclosed something in quotes – then the characters remain in lower case. I found this convention disconcerting and confusing at times. This might not be the case for someone who has not used other PC systems.

The trace facility for debugging of programs is rudimentary and nowhere near as powerful and convenient to use as the Sharp PC's capabilities.

I don't like not being able to save programs and data on a mass storage device such as a tape cassette. Apparently no such devices are available for the Panasonic at this time. You can copy files from one memory bank to another. These banks are battery powered and the batteries can even be changed (quickly!) without losing the contents of memory. However, at a list price of $325.00 for an 8K module, I don't want to use them for archive purposes!

It also appears that once you leave the BASIC operating system, all the variable values assigned to a BASIC program are lost. Going back, for instance, to check an appointment on the built-in appointment organizer, means terminating all calculations and values in whatever BASIC program you might be running.

Finally, there appears to be a slight bug in the BASIC interpreter on the unit I have been using. If you have a Panasonic HHC, try entering the following statements using the Microsoft BASIC capsule:

```
10 X=1.23:X$=STR$(X):X1=VAL(X$):PRINT X1
```

Execute the sequence and let me know if you get a strange result. Maybe we have something that should be reported to **Friends Amis.**

So what is the bottom line? I don't feel this is a particularly good unit for people who just want to putter around with a PC using BASIC. The machine does not appear to be well suited for this purpose. Pick something a little "friendlier" and less expensive. But, am I saying the Panasonic HHC is a bad unit? Nope, not at all. It is just that it is designed for certain applications more than others!

Where It Shines

This unit seems best suited for applications where the software has been designed by skilled professionals to operate in essentially a "canned" mode. Such programs can be provided on ROM capsules that are simply plugged into the back of the unit. An example of one such application was demonstrated recently to me by *Dr. Brown* of **American Medical Instruments,** *PO Box 1080, El Cerrito, CA 94530,* telephone (415) 525-1113. This firm has a program supplied on a ROM capsule that determines the correct I.V. (intraveineous) drip rate for a variety of commonly used drugs based on the patient's body weight, desired dosage, etc. The program operates in a variety of modes. It can serve as a drug-determining calculator to provide the proper drip rate. It can be placed in a mode where it actually issues audible beeps so that the I.V. drip may be synchronized to a specific rate. Or, an operator can press a key each time a drop comes out of the I.V. and the unit will display the rate, etc. Presto, you have an application that is admirably suited for the machine. Complex calculations, often performed at times of stress (medical emergencies are not picnics for doctors and nurses), can now be performed without error using the proper dosages for each type of drug (as stored in the PC's memory).

The fact of the matter is that the Panasonic/Quasar HHC is really a high performance, industrial grade machine. It is designed to be customized to specific applications by professional software developers. The unit is fast. In a test I conducted using Microsoft BASIC, running a program that performed a wide variety of operations from string handling to number crunching, the Panasonic HHC was about three times faster than the equivalent program on a Sharp PC-1500. BASIC programs can be compiled to run even faster using a BASIC – SNAP translator about to be made available for the machine. In critical arenas and commercial or industrial environments this significant speed advantage may well be worth the price.

Friends Amis, *505 Beach Street, San Francisco, CA 94133,* the HHC's system software developer, has a version of the SNAP compiler available for commercial software developers that runs on an Apple II computer. Programs for the HHC can thus be developed and programmable ROMs burned in an environment that is more efficient than using the HHC itself to develop sophisticated software.

Panasonic is reportedly moving full steam ahead on developing additional peripherals to tie-in with the HHC. A four-color printer/plotter unit is reported to be scheduled for release this summer with a retail price in the vicinity of $500.00. A 160K-byte 3-inch diskette cartridge unit is apparently in the works with initial deliveries due near the end of the year.

If you have a professional or business application and need a high-performance, quality HHC, take a good look at this unit. Software houses looking for a system conducive to being set up for specific applications might also be well advised to look the machine over carefully. The investment necessary to produce ROMs for serious applications will probably reduce the competition – maybe even keep it at a level where some people can turn a profit!

PC-1500 Tid-Bits

A lot of readers are hard at work analyzing the ROM code. It would be nice if key routines that could be used as subroutine calls could be located and identified. The recently released *Sharp PC-1500 Service Manual* does not contain information related to the ROM or even give the instruction set of the customized CPU. However, it does indicate

A UNIQUE SYSTEM ORGANIZER

Designed for the Sharp PC-1211 or Radio Shack TRS-80 Pocket Computer (PC-1) equipped with the printer/cassette interface, this *P/C Desk Console* has room for three cassettes, storage of 3 by 5 inch index cards, two rolls of paper and a spare printer ribbon as well as the cassette interface cables. It all measures just 8-1/2 by 16 by 2-3/4 inches ready to keep your system organized and easy to use. The price is $19.95 (plus $2.50 s/h and 6% tax if delivery is in California). Order from Fox/Walker, *4650 Arrow Highway, Building G-17, Montclair, CA 91763.* Tell 'em you saw it in *PCN.*

that the CPU contains the following registers: a 16-bit program counter, stack pointer and "W" register, an 8-bit accumulator and an 8-bit "E" register (probably for flags or extension of the accumulator), and six more 8-bit registers UH and UL, XH and XL and YH and YL that are general purpose and can apparently be paired to form 16-bit data (pointer) registers.

The consensus amongst several experimenters so far is that the machine code BE represents a subroutine call and is followed by two bytes containing the absolute address (high then low address bytes). The code E2 is likely a subroutine return directive. The code AE seems to be "store the accumulator" at the address shown in the next two bytes. B5 apparently is a "load the accumulator" with the data in the next byte.

With what is now known it should be possible to eventually ascertain the machine code instruction set for this device. The key is to share your findings so that others can build upon earlier findings.

Lots of people are sending in information about undocumented capabilities. It is a little too early to attempt to summarize or publish all this material, but take notes and keep us informed of your discoveries. *Thomas S. Cox* has apparently figured one feature out that had a number of users climbing the walls. His words are a lot more informative than the manual:

"Merging several programs having the same line numbers is possible if the programs are labeled and either the RESERVE or DEF modes are used to access the programs. When using this method, the programs can be run with no problems. I should emphasize that the labels must be present in the tape version of the program as it is not possible to edit the first programs in a series of MERGED programs unless the programs are entered such that the line numbers are in ascending order as the programs are loaded. By using non-conflicting labels, it is possible to merge and run programs without having to go to the trouble of renumbering the programs and entering them in precise order." – Thanks, Thomas, that is a nice clarification.

There is other information of this nature coming in and I will try to sort it out and get it disseminated to you as soon as practical.

This and That

Roland J. Saam sends word from England that Sharp has a new product for the PC-1500 designated the CE-153. It is called a "Software Board." It is really a big matrix of 140 keys arranged as 10 rows in 14 columns. You can place a template over it and use software to define each key.

The CE-155 8K RAM module is available now in the United States. I have one installed and it sure is nice to have 10K available on my PC. The list price is $150.00. Shop around.

Sharp is beginning to advertise the PC-1500 in some of the computer and electronic trade journals. So is Panasonic. What took them so long?

Nanos Systems Corporation, *PO Box 24344, Speedway, IN 46224,* has introduced the *TRS-80 BASIC Pocket Computer System Reference Card* (which also is suitable for the Sharp PC-1211). The price is $2.95 (plus 4% tax in Indiana). This is a reference card that folds to fit in a pocket. Expanded it measures 19 by 8-1/2 inches and is printed on both sides with information such as instructions, error codes, system limitations and so forth. Might be especially suitable for beginning programmers.

Radio Shack has announced the release of *Games II* for the TRS-80 Pocket Computer. The package includes eight games on two cassettes plus an instruction manual. The games are: Missile Marksman, Baccarat, Blackjack, Aceydeucey, One-Armed Bandit, Pokerslot, Numguess and Craps. The list price is $14.95.

If the review of the Casio FX-702P in Issue 09 of *PCN* wasn't enough for you, *Bob McElwain* has written a splendidly thorough review of the unit and its capabilities in the May, 1982 issue of **Interface Age.** It starts on page 88 therein. Read it if you have any thoughts about obtaining the FX-702P.

Remember, the next issue of *PCN* will not be out until July. But, there is a good chance there will be a special **Advertisers Edition** in the interim and we are planning a few surprises for the July edition.

Help keep your fellow readers informed by keeping us informed.

– Nat Wadsworth, Editor

SUBSCRIPTION INFORMATION

Subscriptions are sold on a calendar year basis (January – December). There are ten issues per year. You get all issues published for the calendar year to which you subscribe.

☐ 1982 Regular Subscriber (Issues 11 – 20). $30.00 in U.S. (U.S. $36.00 to Canada. U.S. $45.00 elsewhere.)

☐ Sample issue. $3.00 in U.S. (U.S. $4.00 elsewhere.) *Due to credit card minimum, this item cannot be charged.

• • • •

Orders must be accompanied by payment in full. We do not issue invoices for the POCKET COMPUTER NEWSLETTER. Thank you for your remittance.

Name:_____

Addr: _____

City: _____ State: ___ Zip: _____

MC/VISA # _____ Expires: _____

Signature: _____

35 Old State Rd, Oxford, CT 06483